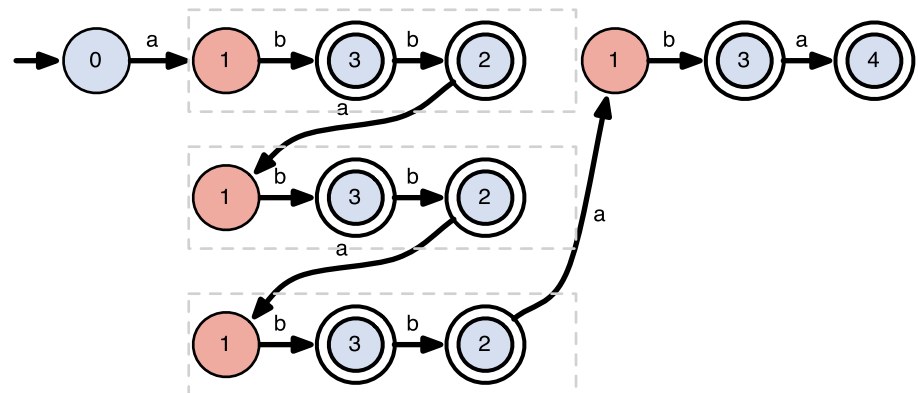


Berechenbarkeitstheorie

3. Vorlesung



Dr. Franziska Jahnke

Institut für Mathematische Logik und Grundlagenforschung

WWU Münster

Definition

Ein **Nichtdeterministischer Endlicher Automat (NEA)**

ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$, mit:

- $|Q| < \infty, |\Sigma| < \infty,$
- $\delta: Q \times \Sigma_\varepsilon \rightarrow \mathcal{P}(Q)$ (möglicherweise partiell!),
- $q_0 \in Q, F \subseteq Q.$

Ein Wort kann verschiedene Läufe haben !!

Akzeptanz eines Wortes (NEA)

Ein NEA N **akzeptiert ein Wort** $w \in \Sigma^*$ genau dann, wenn **ein** w -Lauf von N in einem akzeptierenden Zustand endet.

Erkennen einer Sprache (NEA)

Ein NEA N **erkennt (akzeptiert) die Sprache**

$$L(N) := \{w \in \Sigma^* \mid N \text{ akzeptiert } w\}.$$

Iterierter Übergang beim NEA

3

3. Vorlesung

ε -Abschluss

(wo komme ich hin ohne ein Zeichen zu lesen?)

$E(p \in Q) := \{q \in Q \mid q \text{ über } \geq 0 \text{ } \varepsilon\text{-Kanten von } p \text{ erreichbar}\}$

$E(P \subseteq Q) := \bigcup_{p \in P} E(p)$

Definition

Für einen NEA $N = (Q, \Sigma, \delta, q_0, F)$ ist die **iterierte**

Übergangsfunktion $\delta^k : Q \times \Sigma^k \rightarrow \mathcal{P}(Q)$ definiert durch:

1. $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
2. $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q, w)} \delta(r, x)\right)$ für alle $q \in Q$,
 $w \in \Sigma^{i-1}, x \in \Sigma$.

Schreibweise: $\delta^*(q, w)$ für $\delta^{|w|}(q, w)$

Iterierter Übergang beim NEA

3

3. Vorlesung

ε -Abschluss

(wo komme ich hin ohne ein Zeichen zu lesen?)

$$L(N) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \cap F \neq \emptyset\}$$

$$E(p) = \bigcup_{p \in F} \{p\}$$

Definition

Für einen NEA $N = (Q, \Sigma, \delta, q_0, F)$ ist die **iterierte Übergangsfunktion** $\delta^k : Q \times \Sigma^k \rightarrow \mathcal{P}(Q)$ definiert durch:

1. $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$,
2. $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q, w)} \delta(r, x)\right)$ für alle $q \in Q$,
 $w \in \Sigma^{i-1}$, $x \in \Sigma$.

Schreibweise: $\delta^*(q, w)$ für $\delta^{|w|}(q, w)$

Satz 2

$$\{L \mid L \text{ wird durch NEA erkannt}\} = \text{REG}$$

Potenzautomat

Für einen NEA $N = (Q, \Sigma, \delta, q_0, F)$ ist sein

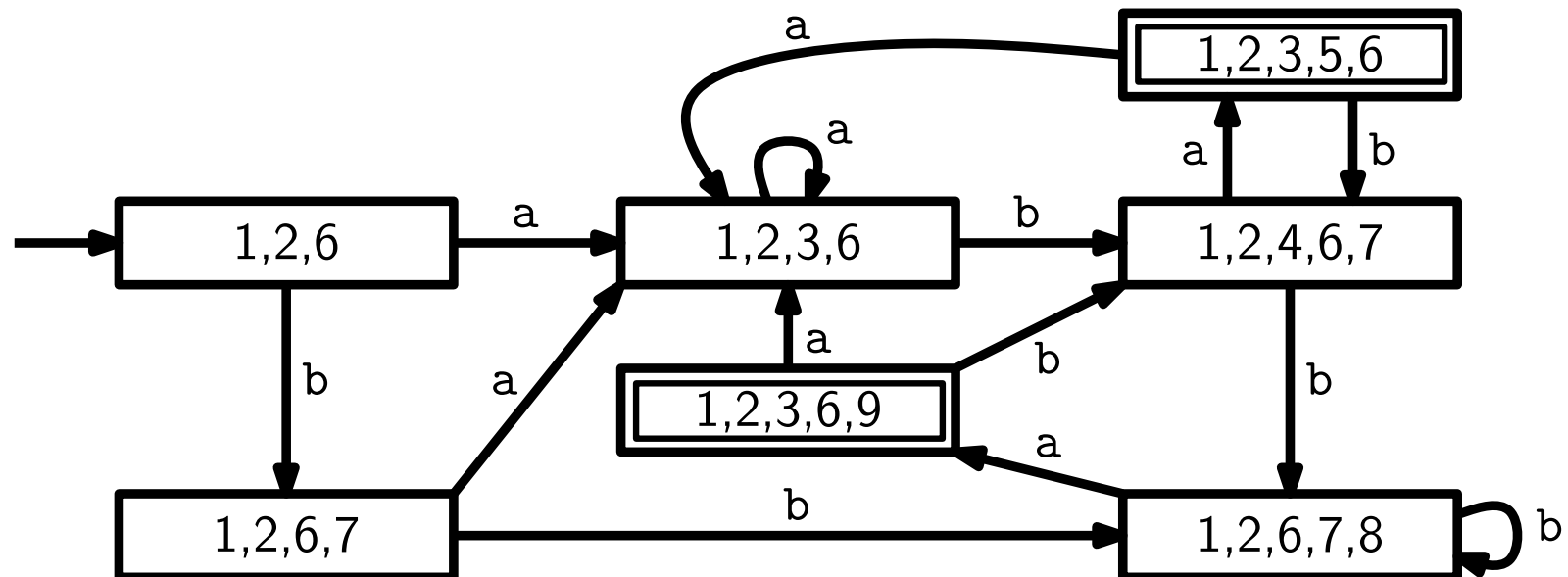
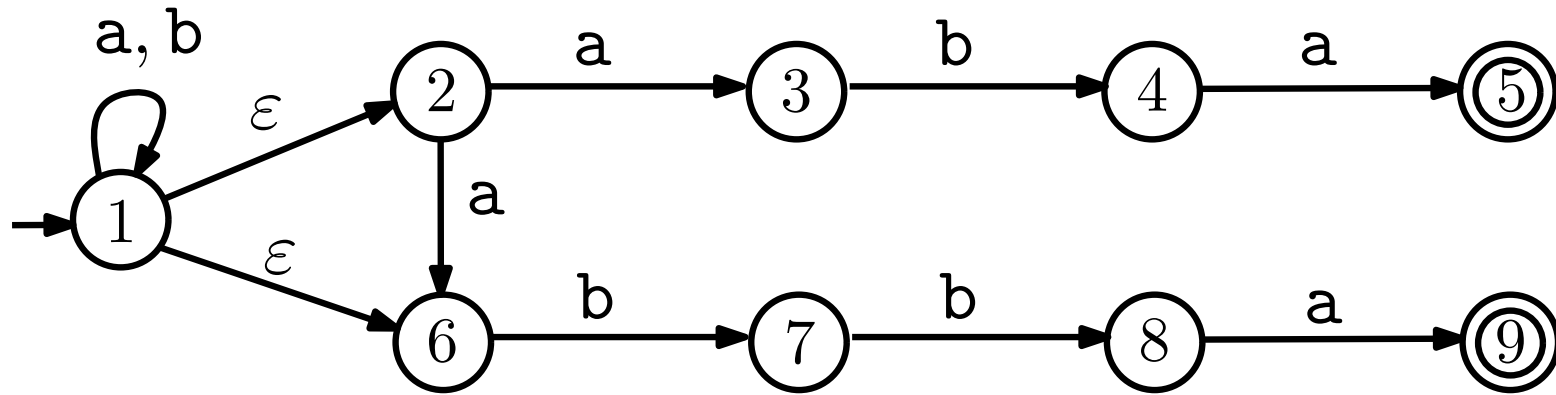
Potenzautomat $\text{Pot}(N) = (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{q}_0, \tilde{F})$ ein DEA mit

- Zustandsmenge $\tilde{Q} = \mathcal{P}(Q)$,
- Übergangsfunktion $\tilde{\delta}$, mit $\tilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$,
- Startzustand $\tilde{q}_0 = E(q_0)$,
- akzeptierenden Zuständen $\tilde{F} = \{P \in \tilde{Q} \mid P \cap F \neq \emptyset\}$.

Behauptung

$$L(N) = L(\text{Pot}(N))$$

Beispiel Potenzautomat



Lemma 1: $\tilde{\delta}^*(E(q), w) = \delta^*(q, w)$

Beweis durch vollständige Induktion über $|w|$.

Ind.anfang: $\tilde{\delta}^0(E(q), \varepsilon) = E(q) = \delta^0(q, \varepsilon)$

Ind.schritt: $\tilde{\delta}^{k+1}(E(q), ux) = \tilde{\delta}(\tilde{\delta}^k(E(q), u), x)$ für $w = ux, x \in \Sigma$
 $= \tilde{\delta}(\delta^k(q, u), x)$ Induktionsannahme

$$= E\left(\bigcup_{p \in \delta^k(q, u)} \delta(p, x)\right) \quad (2)$$

$$= \delta^{k+1}(q, ux) \quad (4)$$

□

Erinnerung:

- (1) $\tilde{\delta}^0(P, \varepsilon) = P$ für alle $P \in \tilde{Q}$, ← Def. δ^0 für DEA
- (2) $\tilde{\delta}(P, x) = E(\bigcup_{p \in P} \delta(p, x))$ für alle $P \in \tilde{Q}$, ← Def. $\tilde{\delta}$
- (3) $\delta^0(q, \varepsilon) = E(q)$ für alle $q \in Q$, ← Def. δ^i für NEA
- (4) $\delta^i(q, wx) = E\left(\bigcup_{r \in \delta^{i-1}(q, w)} \delta(r, x)\right)$ für alle $q \in Q, w \in \Sigma^{i-1}, x \in \Sigma$. ←

Beweis Satz 2

Wir müssen zeigen $L(N) = L(\text{Pot}(N))$ für jeden NEA N

$$\begin{aligned}
 w \in \Sigma^* \text{ wird von } N \text{ akzeptiert} &\Leftrightarrow \delta^*(q_0, w) \cap F \neq \emptyset \\
 &\Leftrightarrow \tilde{\delta}^*(E(q_0), w) \cap F \neq \emptyset && \text{(nach Lemma 1)} \\
 &\Leftrightarrow \tilde{\delta}^*(\tilde{q}_0, w) \cap F \neq \emptyset && \text{(nach Def. } \tilde{q}_0) \\
 &\Leftrightarrow \tilde{\delta}^*(\tilde{q}_0, w) \in \tilde{F} && \text{(nach Def. } \tilde{F}) \\
 &\Leftrightarrow \text{Pot}(N) \text{ akzeptiert } w
 \end{aligned}$$

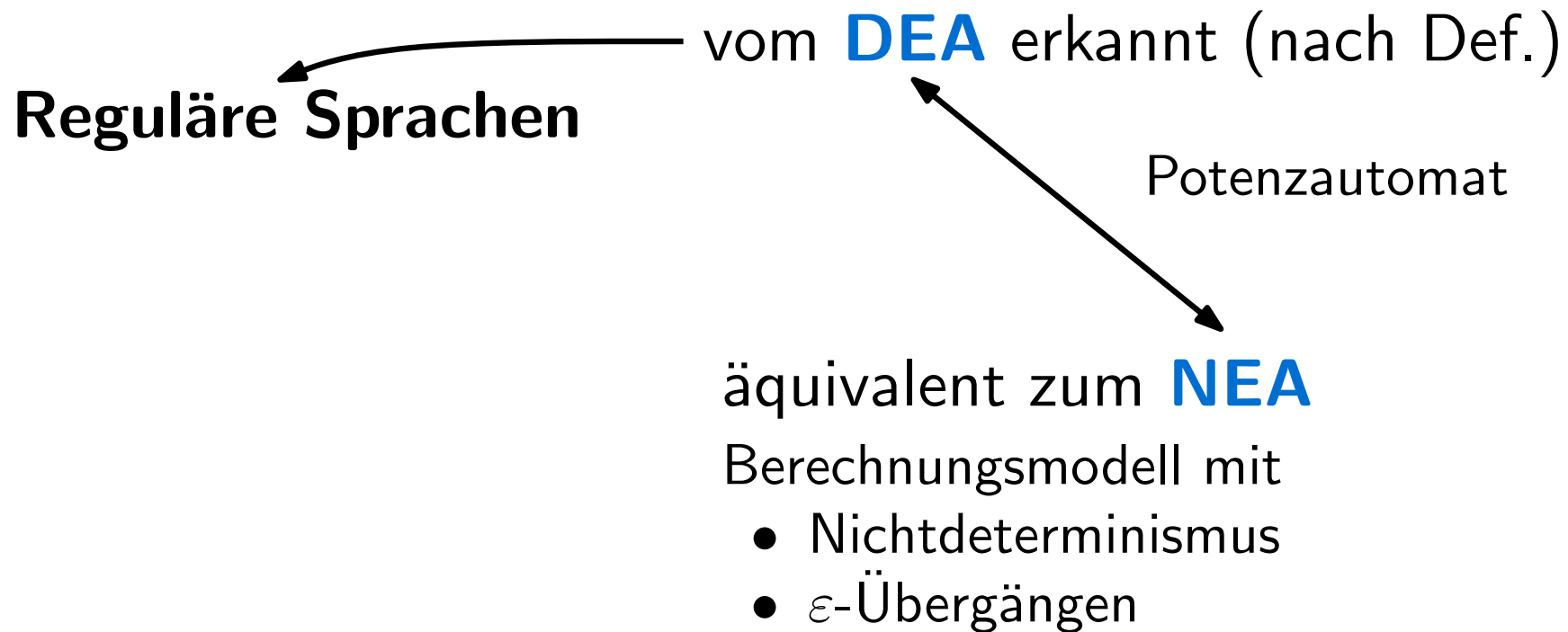
Also:

L wird durch NEA erkannt $\rightarrow L$ wird von DEA erkannt $\rightarrow L \in \text{REG}$

Es folgt:

$$\{L \mid L \text{ wird durch NEA erkannt}\} = \text{REG}$$





Abschlusseigenschaften von REG ⁹

→ die Sprachklasse \mathcal{L} ist abgeschlossen unter der Operation \bowtie , wenn für alle $L_1, L_2 \in \mathcal{L}$ auch $L_1 \bowtie L_2 \in \mathcal{L}$.

Satz 3

Die regulären Sprachen sind abgeschlossen unter binärer Vereinigung.

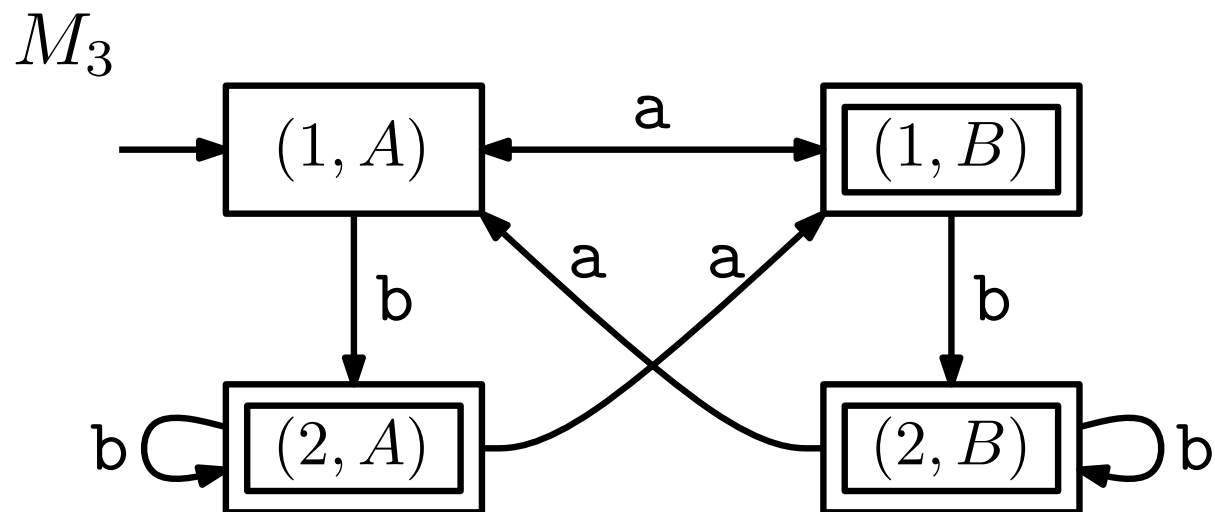
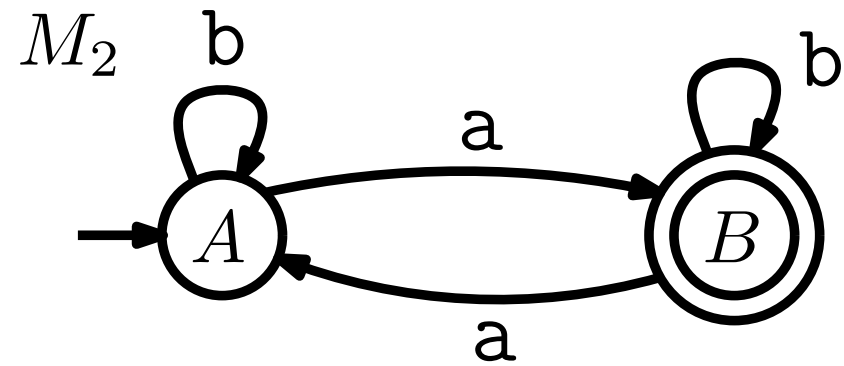
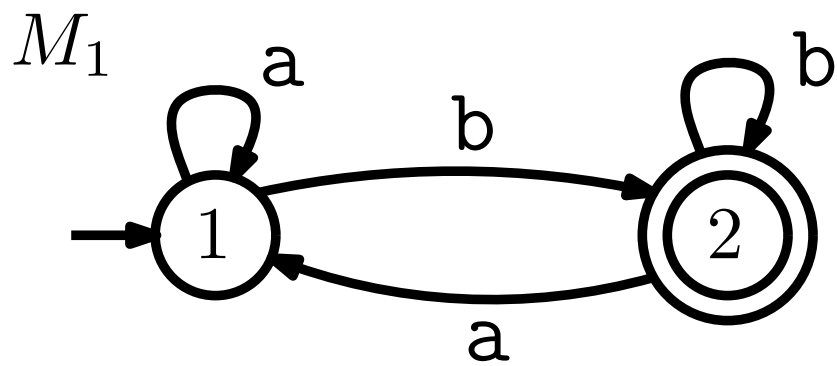
$(L_1, L_2 \in \text{REG} \Rightarrow L_1 \cup L_2 \in \text{REG})$

Beweis 1 für Satz 3

- wenn $L_1, L_2 \in \text{REG}$, dann existieren DEAs M_1 und M_2 mit $L(M_1) = L_1$ und $L(M_2) = L_2$
- Konstruiere DEA M_3 , der M_1 und M_2 simuliert und genau dann akzeptiert, wenn einer der beiden DEAs akzeptiert

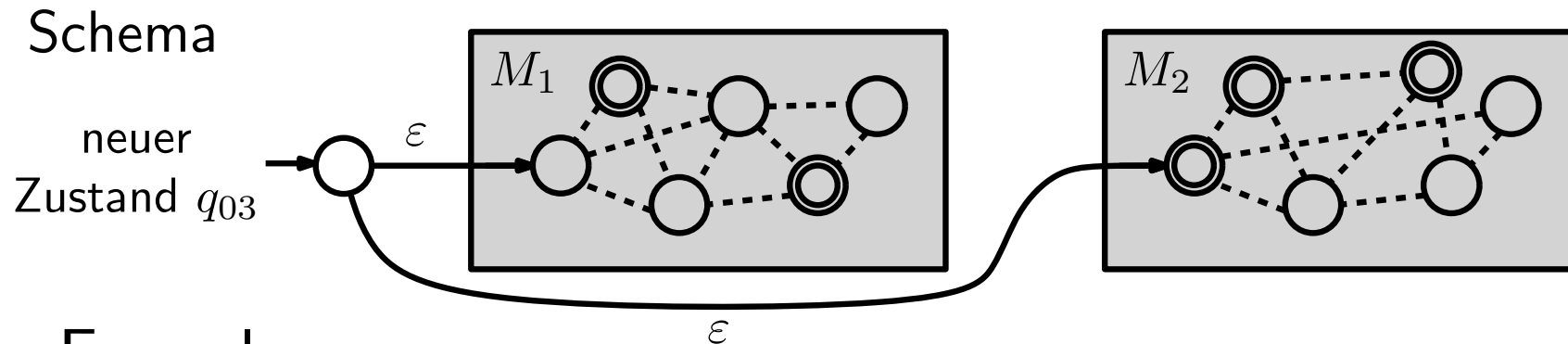
- sei $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ & $M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$
- wir konstruieren $M_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$
- **Idee:**
Zustände von M_3 sind Paare von Zuständen $Q_1 \times Q_2$
→ M_3 ist genau dann im Zustand (q_a, q_b) wenn M_1 im Zustand q_a und M_2 im Zustand q_b sind (Invariante)
- **Formal:**
 - $Q_3 := Q_1 \times Q_2$,
 - $\delta_3((q_1, q_2), a) := (q'_1, q'_2)$, g.d.w. $\delta_1(q_1, a) = q'_1$ und $\delta_2(q_2, a) = q'_2$
 - $q_{03} := (q_{01}, q_{02})$ → garantiert Invariante (ohne formalen Beweis)
- **Akz. Zustände:** akzeptiere, wenn M_1 oder M_2 akzeptiert
 - $F_3 := \{(q_1, q_2) \mid q_1 \in F_1 \text{ oder } q_2 \in F_2\}$ □

Beispiel



Beweis 2 für Satz 3

- M_1, M_2 wie bisher, wir konstruieren nun **NEA**
 $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \cup L_2$



- Formal:

- $Q_3 := Q_1 \cup Q_2 \cup \{q_{03}\},$

- $\delta_3(q, a) := \begin{cases} \{q_{01}, q_{02}\} & \text{wenn } q = q_{03} \text{ und } a = \varepsilon \\ \emptyset & \text{wenn } q = q_{03} \text{ und } a \neq \varepsilon \\ \delta_1(q, a) & \text{wenn } q \in Q_1 \\ \delta_2(q, a) & \text{wenn } q \in Q_2 \end{cases}$

- $F_3 := F_1 \cup F_2$

Annahme:

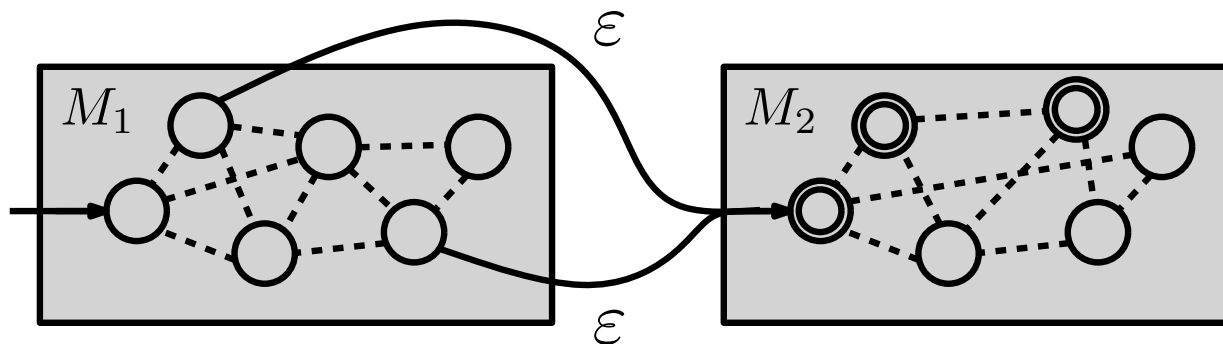
Q_1 und Q_2 disjunkt
 $q_{03} \notin Q_1 \cup Q_2$

Satz 4

Die Regulären Sprachen sind abgeschlossen unter Konkatenation, Kleene Stern und Komplement.

Beweisskizzen

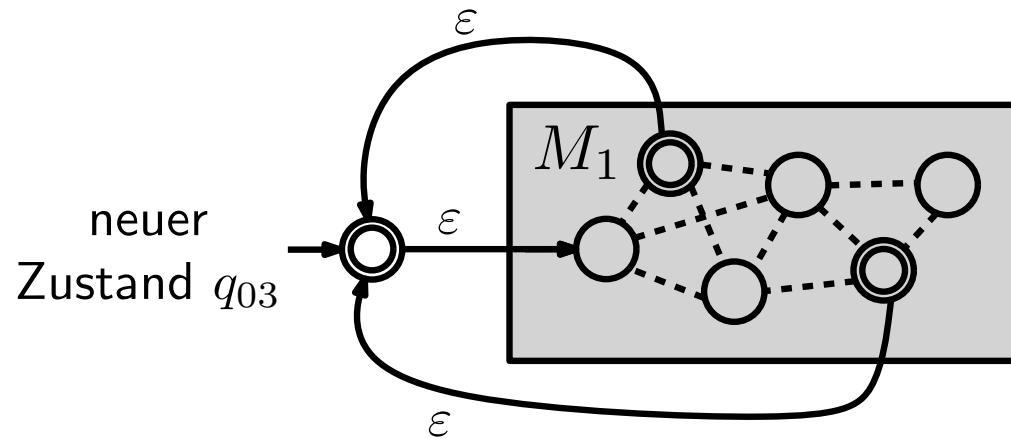
- M_1, M_2 wie bisher, wir konstruieren nun **NEA**
 $N_3 = (Q_3, \Sigma, \delta_3, q_{03}, F_3)$ für $L_1 \circ L_2$



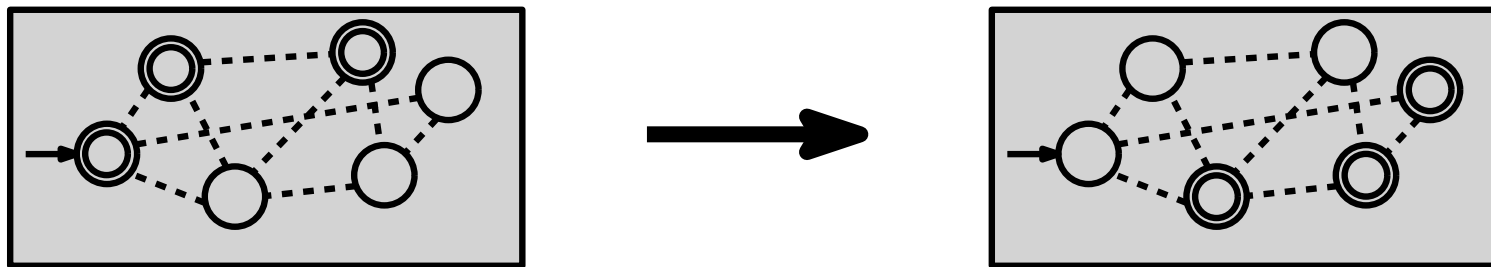
- verbinde alle akz. Zustände von M_1 über ε Kanten mit q_{02} und setze $q_{03} = q_{01}$ und $F_3 = F_2$.

Abschluss Kleene Stern und Komplement¹⁴

- M_1 wie bisher, wir konstruieren nun **NEA** N_3 für L_1^*



- neuer Startzustand q_{03} über ϵ -Kante mit q_{01} verbunden
 - neuer Startzustand ist akzeptierend, da $\epsilon \in L(M_1)$
 - verbinde alle Zustände aus F_1 über ϵ -Kanten mit q_{03}
- für **Komplement** tausche akzeptierende Zustände aus



Reguläre Ausdrücke

- Formalismus zum Beschreiben von formalen Sprachen
- ein **Regulärer Ausdruck (RA)** ist ein Wort über dem Alphabet $\Sigma \cup \{\emptyset, \varepsilon, (,), +, \cdot, *\}$

- Syntax**
- (1) $\forall a \in \Sigma \cup \{\emptyset, \varepsilon\}$: a ist RA
 - (2) Wenn A, B Reg. Ausdrücke, dann auch $(A + B)$
 - (3) Wenn A, B Reg. Ausdrücke, dann auch $A \cdot B$ (kurz AB)
 - (4) Wenn A Reg. Ausdruck, dann auch A^*
 - (5) Wenn A Reg. Ausdruck, dann auch (A)

- Bsp**
- $((aba^* + abb)^* ab + \varepsilon)$ ist RA
 - $(a + b)b(*b)$ ist kein RA

Induktive Definition korrekt, da RA durch die Anwendung der (umgekehrten) Regeln immer kürzer werden

Für jeden RA R definieren wir induktiv eine Sprache $L(R)$:

- (1)
 - $R = \emptyset$, dann $L(R) = \emptyset$
 - $R = \varepsilon$, dann $L(R) = \{\varepsilon\}$
 - $R = a$, und $a \in \Sigma$, dann $L(R) = \{a\}$
- (2) $R = (A + B)$ und A, B sind RA, dann $L(R) = L(A) \cup L(B)$
- (3) $R = (AB)$ und A, B sind RA, dann $L(R) = L(A) \circ L(B)$
- (4) $R = A^*$ und A ist RA, dann $L(R) = L(A)^*$
- (5) $R = (A)$ und A ist RA, dann $L(R) = L(A)$

Reihenfolge der Operatoren: $*$ vor \cdot vor $+$

Bsp. $R = (a + b)^* aba(a + b)^*$

$L(R) = \{w \in \{a, b\}^* \mid w \text{ enthält Teilwort } aba\}$

Satz 5

$$\{L \mid \exists \text{ RA } R \text{ mit } L(R) = L\} = \text{REG}$$

Beweis (Organisation)

1. Teil: Wir konstruieren zu einem RA R einen NEA N mit $L(R) = L(N)$

das zeigt $\{L \mid \exists \text{ RA } R \text{ mit } L(R) = L\} \subseteq \text{REG}$

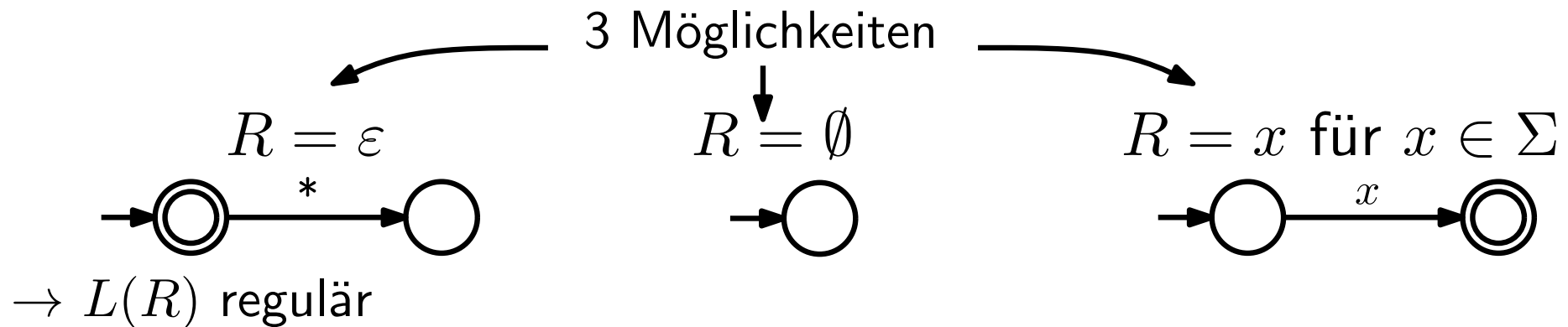
2. Teil: Wir konstruieren zu einem DEA M einen RA R mit $L(M) = L(R)$

das zeigt $\{L \mid \exists \text{ RA } R \text{ mit } L(R) = L\} \supseteq \text{REG}$

Für RA R ist $L(R)$ regulär

Vollständige Induktion über Anzahl der Ableitungsregeln

Ind.anfang: Eine Ableitungsregel

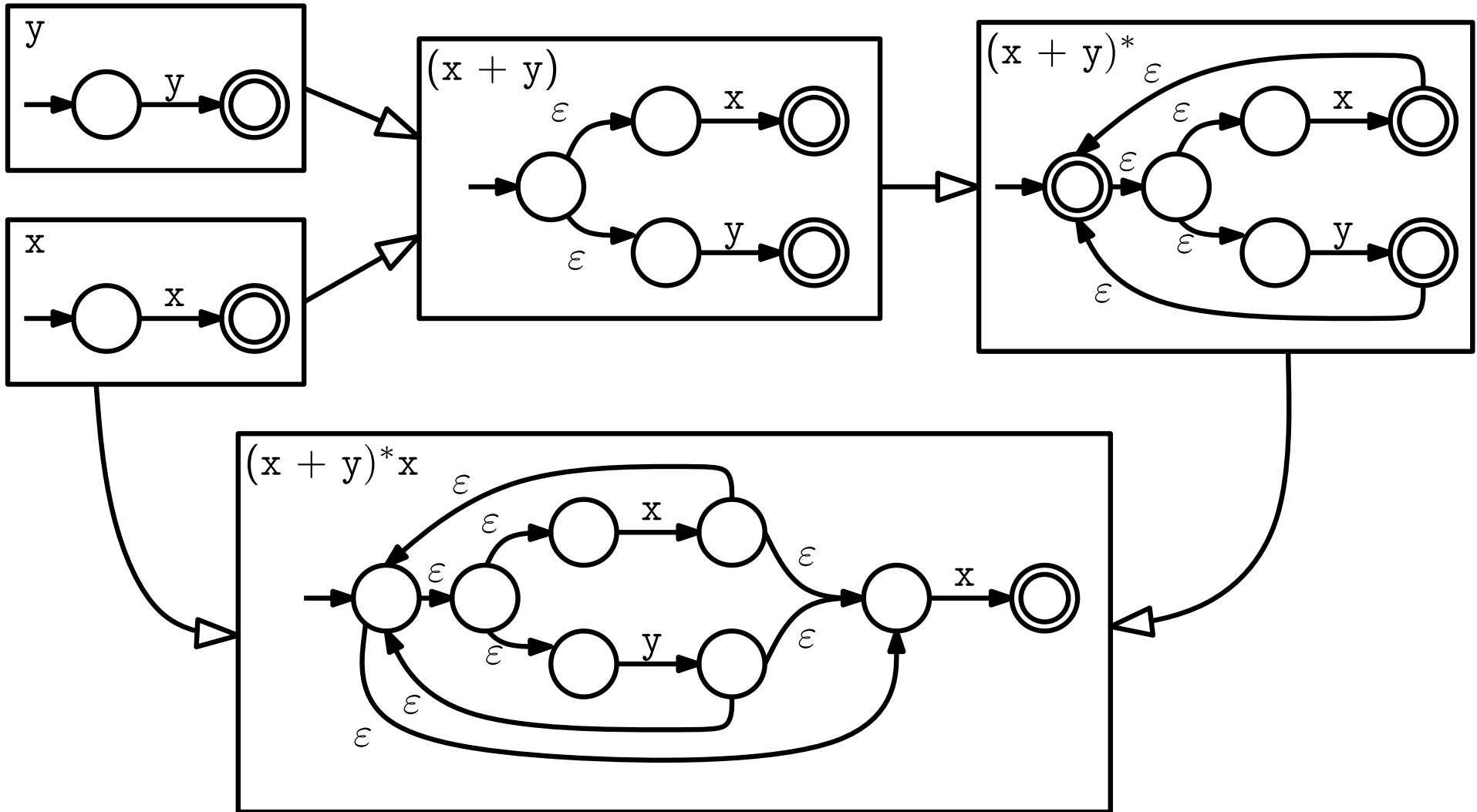


Ind.schritt: k Ableitungsregeln

- Angenommen $R = (A + B)$, und A und B sind RAs.
- Nach Induktionsannahme sind $L(A)$ und $L(B)$ regulär.
- $L(R) = L(A) \cup L(B)$ ist nach Abschlusseigenschaft (Satz 4) auch regulär.
- für Konkatination, Stern, Klammern, analog □

Beispiel

$$R = (x + y)^* x$$

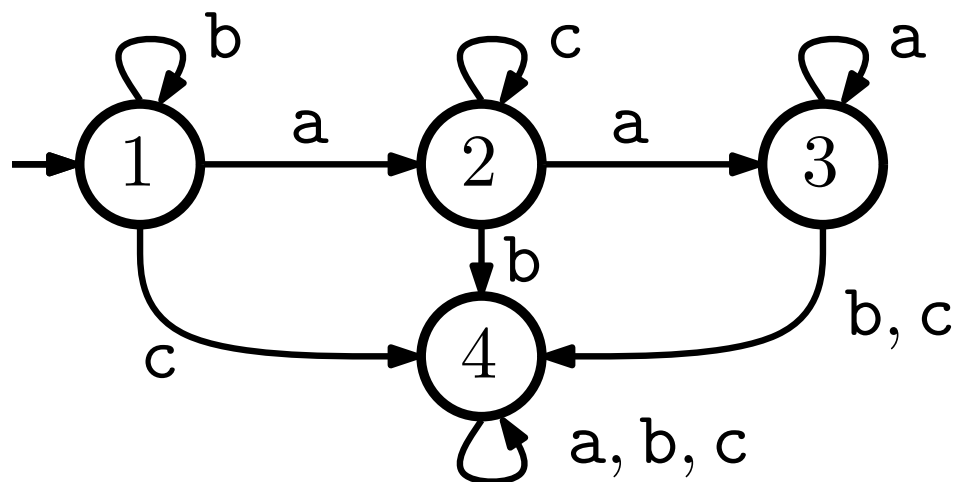


Vom DEA zum RA

- Ansatz**
- Wenn $L \in \text{REG}$, dann existiert ein DEA M der L erkennt.
 - $M = (Q, \Sigma, \delta, 1, F)$ mit $Q = \{1, 2, \dots, n\}$
 - **Ziel:** Konstruiere RA R mit $L(R) = L$

R_{ij}^k := RA für alle Wörter die von Zustand i nach j führen ohne einen Zustand $> k$ zu benutzen.

Anfangs- und Endzustand i, j dürfen $> k$ sein!



Bsp $R_{22}^0 = c + \varepsilon$

$$R_{24}^2 = c^*b$$

$$R_{24}^3 = (c^*b + c^*aa^*(b + c))$$

Annahme: Ich kenne alle Terme R_{ij}^k

→ $w \in L(M)$ genau dann, wenn $\delta^*(1, w) = j$ und $j \in F$

$w \in R_{1j}^n$

oberer Index n heißt keine Beschränkung

→ verbinde für alle akzeptierende Zustände die R_{ij}^k Terme

$$R := R_{1f_1}^n + R_{1f_2}^n + \cdots + R_{1f_k}^n$$

$$\text{wobei } F = \{f_1, f_2, \dots, f_k\}$$

Konstruktion der R_{ij}^k Terme

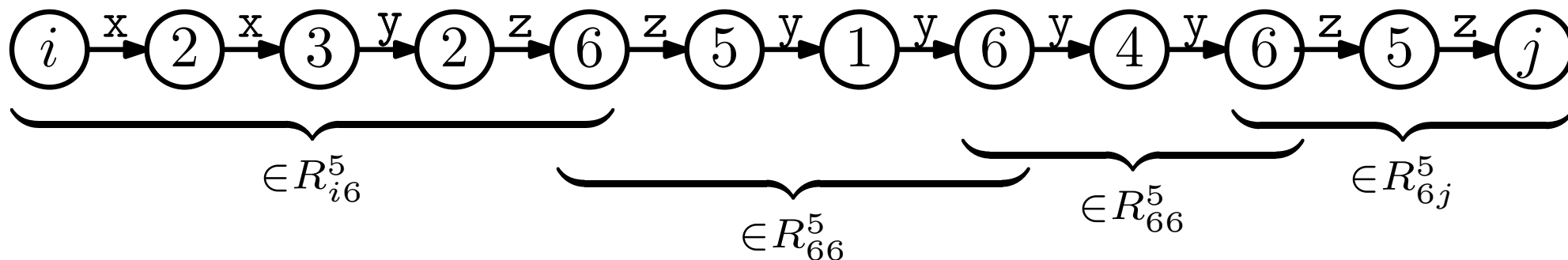
Rekursive Konstruktion

Basisfall (R_{ij}^0): nur direkte Übergänge

$$R_{ij}^0 := \begin{cases} \emptyset + a_1 + a_2 + \dots & \text{falls } i \neq j \text{ und } a_i \in \{a \mid \delta(i, a) = j\} \\ \varepsilon + a_1 + a_2 + \dots & \text{falls } i = j \text{ und } a_i \in \{a \mid \delta(i, a) = i\} \end{cases}$$

Rekursion (R_{ij}^k): ich kenne bereits alle R_{ij}^{k-1}

Bsp. von einem "Lauf" aus R_{ij}^6



$\rightarrow R_{**}^k$ kann ich mit Termen R_{**}^{k-1} beschreiben

Rekursion für R_{ij}^k

- 1. Möglichkeit: der *Lauf* eines Wortes besucht **nicht** den Zustand k
 $\rightarrow R_{ij}^k \supseteq R_{ij}^{k-1}$
- 2. Möglichkeit: der *Lauf* eines Wortes besucht einmal oder mehrmals den Zustand k
 $\rightarrow R_{ij}^k \supseteq R_{ik}^{k-1} (R_{kk}^{k-1})^* R_{kj}^{k-1}$
- es gibt keine andere Möglichkeit, deshalb

$$R_{ij}^k = R_{ij}^{k-1} + R_{ik}^{k-1} (R_{kk}^{k-1})^* R_{kj}^{k-1}$$

Berechnung mit ansteigendem k ($k = 0, \dots, n$) aller R_{ij}^k Terme

Dynamisches Programmieren

